DeBruijn Strings, Double Helices, and the Ehrenfeucht-Mycielski Mechanism

Terry R. McConnell
Syracuse University

April 16, 2013

Abstract

We revisit the pseudo-random sequence introduced by Ehrenfeucht and Mycielski and its connections with DeBruijn strings.

1 Introduction

DeBruijn strings of order n are finite strings of binary digits of minimal length that contain as substrings all 2^n possible binary strings of length n. Since the addition of a single binary digit can produce at most one new substring of length n, it is easy to see that a DeBruijn string must have length at least $2^n + n - 1$. It is remarkable that this minimum is achieved - it is possible to pack the greatest possible variety of binary strings of given length into the minimum possible space.

For example, the string 0011101000 is a DeBruijn string of order 3, and the string 1111011001010000111 is a DeBruijn string of order 4. It follows from classical work of DeBruijn that there are exactly $2^{2^{n-1}}$ distinct DeBruijn strings of order n. (See, e.g., p. 136 of [4].) The would-be DeBruijn-string enthusiast can do no better than to begin with the excellent survey [3] of Fredrickson.

Here is a simple procedure to generate an example of a DeBruijn string of any desired order n: begin with an n-string that is all zeros except for a final 1. At each subsequent stage the next digit is a 1 unless that would cause a repeat of the terminal n-string. Otherwise, it is a zero. Continuing in this way will always produce a DeBruijn string that ends with n zeros. (For a proof, see, e.g., pages 4-6 of [7], where the algorithm was naively presented as new. In fact, as noted in [3], the algorithm is very well-known, often rediscovered, and goes back at least to 1934. It is known in the literature as the 'prefer-one algorithm'.)

The Ehrenfeucht-Mycielski (hereafter EM) sequence is an infinite binary sequence with pseudo-random properties first introduced in [1]. The sequence begins 010, and thereafter each additional digit is generated by an algorithm that attempts strenuously to avoid repetition. The precise algorithm is as follows: Find the longest suffix of the sequence generated so far that occurred at least once previously. The next digit is then the opposite of the one that followed the penultimate occurrence of that suffix. Thus, the first 15 digits of EM are 010011010111000. It is easy to program a computer to calculate long prefixes of EM. For example, the first million digits are available at [2].

Since its introduction in 1992, the properties of EM have been studied by several authors. See, e.g., [7],[9],[6], and [5]. It is sequence number A007061 in the Sloane On-line Sequence Encyclopedia [8].

The EM and prefer-one algorithms are similar in that both seek to avoid repetition. The main goal of this paper is to explore connections between the EM sequence and DeBruijn strings. For example, we show in section 3 below that if the comparison of binary strings in the EM algorithm is done in a finite buffer of fixed length n, then the resulting sequence will eventually become periodic, and that the repeating unit is a DeBruijn string of order $n+1$. This leads to a simple-minded and efficient algorithm for generating all possible DeBruijn strings of a given order. The algorithm may be new, albeit rather closely related to ALGORITHM 1 of [3].

We also consider seeded variations of EM, and show, in particular, that the sequence uniquely determines the seed.

DeBruijn strings have many connections to other fields, including graph theory, coding theory, and shift-register sequences. For example, a DeBruijn string may be viewed as a Hamiltonian circuit of the DeBruijn graph of the same order. (See the following section for the relevant definitions.) If the edges of such a circuit are removed from the graph, the remaining graph always has a least 3 connected components, including simple loops on the vertices corresponding to the all-zero and the all-one string. In cases where there is only a single remaining (long loop) component, we term the original DeBruijn string a *double helix.* In the last section of the paper we show that double helices of all orders exist, and consider what happens when a double helix is used as seed for the EM sequence.

2 Terminology and Notation

The subject matter of this paper lies at the intersection of several fields, including Computer Science and parts of Mathematics that study combinatorics of words and free monoids. Accordingly, there is a variety of notation and terminology in common use. For example, we prefer the word ‘string’ to ‘binary sequence’ or ‘binary word’, and will use it in the rest of the paper. Similarly, we prefer to use ‘tail’ in place of ‘suffix’, and ‘head’ in place of ‘prefix’. We consider only strings over the alphabet $\{0,1\}$. It is likely some results can be generalized to larger alphabets, but we have not pursued this.

We shall use either small Greek letters for strings, or else capital Roman ones. The distinction is that Greek letters will be for ‘short’ strings, and the others for ‘long’ ones. String concatenation is indicated by juxtaposition of symbols. Individual binary digits will be denoted by x, y, z, etc. A prime indicates complementary binary digit, i.e., if $x = 0$, then $x' = 1$, and *vice versa.* The set of all binary strings is denoted Σ.

We denote by σ_- the head, and by $_-\sigma$ the tail, of the string σ. More precisely, if $\sigma = x\tau y$ for some string τ and digits x and y, then $\sigma_- = x\tau$, and $_-\sigma = \tau y$. The length of σ is $|\sigma|$. If σ is a string of length at least n, then σ_n shall denote the head of σ of length n. The symbols $\bar{0}_n$ and $\bar{1}_n$ indicate strings of length n comprised of all 0s and all 1s respectively.

It will be convenient to use a few terms from [7] in connection with the EM sequence. Given an initial string of EM, the longest tail σ having an earlier occurrence will be called the *match string*, and $|\sigma|$ will be called the *match length.*

The DeBruijn graph of order n is the directed graph $\mathcal{B}^n$ whose vertices are labeled by each of the possible binary strings of length n. If $\sigma = x\tau$ labels a given vertex, then there is an edge connecting from that vertex to each of the vertices labeled $\tau 0$ and $\tau 1$. There are no other edges than these. A given nonempty binary string X of length at least n induces a path on $\mathcal{B}^n$ by starting at the vertex labeled with the initial n-string of X, and following with the vertices labeled by each successive substring of X of length n.

The graph $\mathcal{B}^{n+1}$ is, in a natural way, the dual graph of $\mathcal{B}^n$: Each edge $x\tau \to \tau y$ of $\mathcal{B}^n$ is made to correspond with vertex $x\tau y$ of $\mathcal{B}^{n+1}$. There is an edge in $\mathcal{B}^{n+1}$ between vertices corresponding to two given edges of $\mathcal{B}^n$, if and only if the latter edges meet at a common vertex. Under this correspondence, a given DeBruijn string X of order $n+1$ induces (i) a Hamiltonian path in $\mathcal{B}^{n+1}$; and (ii), an Eulerian path in $\mathcal{B}^n$. (In the first case, every vertex is visited exactly once; and, in the second case, every edge is crossed exactly once.)

3 EM-variants and DeBruijn Sequences

In this section we study certain variants of the EM sequence. First, let us modify the notion of string equality used in the definition of the EM sequence by interposing a transformation. Consider a function $h : \Sigma \to \Sigma$ which maps the empty string to the empty string. We shall deem two strings τ and σ to be *h-equivalent* if $h(\tau) = h(\sigma)$. The EM sequence based on h, denoted EM_h, is the binary sequence $x_1 x_2 \dots$ defined by $x_0 = 0$, and then as follows to define x_{n+1} for $n \geq 0$: Let σ be the longest[1] tail of $x_1 \dots x_n$ for which there is at least one $\tau \in x_1 \dots x_{n-1}$ that is h-equivalent to σ. Then $x_{n+1} = y'$, where y is the follower of the latest such τ.

We also consider sequences with a given nonempty *seed string*, Z. One defines the binary sequence $EM_h(Z)$ by pre-pending Z to the strings $x_1 \dots x_n$ and $x_1 \dots x_{n-1}$ used in the definition of EM_h above. If Z is nonempty then x_1 is determined by the same rule as is used for the later terms. (Note that $EM_h(Z)$ does not include the seed.)

If the function h is one-one then EM_h coincides with the usual Ehrenfeucht-Mycielski sequence, so we confine our attention to functions that are not one-one. A natural class of examples to consider are ones that model testing for

[1] This tail may be empty, in which case the latest previous occurrence is understood to be just before x_n.

string equality in a buffer of fixed finite length. This is, of course, the case in practise when a digital computer with finite memory is used to generate an initial segment of the EM sequence. Thus, given a fixed $n \geq 1$, define the sequence $EM_n(Z)$ (EM_n if the seed is empty) by using for h the function

$$h(\sigma) = \begin{cases} \sigma, & |\sigma| \leq n \\ \sigma_n, & \text{otherwise.} \end{cases} \tag{3.1}$$

The sequence $EM_n(Z)$ is identical to $EM(Z)$ until the match length first exceeds n. Computer experiments show that this initial section is followed by a chaotic stretch of seemingly indeterminate length, after which the sequence becomes periodic. The repeating section is a DeBruijn string of order $n+1$, as we show below.

Proposition 3.1 For $n \geq 2$ a DeBruijn string X of order n contains every string of length $n-1$ with opposite followers. All but the initial and terminal strings (which must be equal) occur exactly twice. The initial string occurs 3 times.

Proof: Let τ be a string of length $n-1$. Since both $\tau 0$ and $\tau 1$ occur in X, τ must occur at least twice. Suppose there where a third occurrence. Then τ must be the initial string of X to avoid a repetition of either 0τ or 1τ. But τ must also be the terminal string of X in order to avoid a repetition of either $\tau 0$ or $\tau 1$. □

If a DeBruijn string Z of order n begins with an n-string σ, then it ends with σ_-. This follows from Proposition 3.1. Closely related to DeBruijn strings are *DeBruijn cycles.* Drop the final σ_- of Z and view the resulting string as a cycle of length 2^n that begins afresh with the beginning σ after 2^n terms. The literature on DeBruijn cycles is somewhat confusing since they have been called various things. For example, such cycles are called *full cycles* in [3].

If each of the 2^n cyclic permutations of a DeBruijn cycle is extended to a DeBruijn string in the obvious way, then 2^n distinct DeBruijn strings are obtained, one for each possible initial σ. It was originally proved by DeBruijn (see, e.g., p. 136 of [4]) that there are exactly $2^{2^{n-1}-n}$ distinct DeBruijn cycles. Thus, there are exactly $2^{2^{n-1}}$ possible DeBruijn strings for a given n. (For $n = 4$, strings yielding 8 distinct DeBruijn cycles are listed on p. 135 of [4]. These, together with their bitwise complements, comprise all 16 DeBruijn cycles of order 4.)

For any given seed string Z, the sequence $EM_n(Z)$ can be generated by a finite state machine, and therefore it must eventually become periodic. Thus, unlike $EM(Z)$, the $EM_n(Z)$ are not transitive.[2] On the other hand, they do contain all possible strings of length $n+1$:

Proposition 3.2 Every string of length $n+1$ occurs infinitely many times in $EM_n(Z)$.

[2] An infinite string is called *transitive* if it contains every possible string of finite length as a substring.

Proof: It is easy to see that $EM_n(Z)$ cannot eventually become constant, and therefore both binary digits occur infinitely many times. On the other hand, since $EM_n(Z)$ is ultimately periodic, there are sufficiently long strings that do not occur at all. Thus there is a shortest length string, τ say, that occurs at most finitely many times. Clearly $|\tau| \geq 2$. Thus τ_- is a nonempty string occurring infinitely many times. Suppose $|\tau_-| \leq n$. Then by the 'pigeonhole principle' there is some string ν (possibly empty) such that $|\nu\tau_-| = n$ and $\nu\tau_-$ occurs infinitely many times. But the formation rule for $EM_n(Z)$ then yields infinitely many occurrences of both τ_-0 and τ_-1. In particular, τ itself occurs infinitely many times, contrary to assumption. Thus it must be the case that $|\tau_-| \geq n + 1$. □

Proposition 3.3 Let X be a DeBruijn string of order $n+1$ with initial n-string σ, say X $= \sigma Y \sigma$. Then $EM_n(X) = Y\sigma Y\sigma \ldots$.

This is an easy consequence of Proposition 3.1.

Theorem 3.1 For any key Z, $EM_n(Z)$ will eventually produce a DeBruijn string of order $n + 1$. After that, the sequence will repeat as if the DeBruijn string had been the key.

Proof: Let τ be a string of length $n+1$. We shall say that a natural number p is a *period* of τ if there are infinitely many non-overlapping substrings X of $EM_n(Z)$ with $|X| = p$, and that begin and end with τ with no other τ in between. Since $EM_n(Z)$ is ultimately periodic, we may choose and fix a string τ having a maximum possible period. Let W denote the (infinite) tail of $EM_n(Z)$ that begins at the earlier of the two occurrences of τ in one such X.

Let σ be a string of length $n + 1$ occurring in W such that the substring extending from the first occurrence of σ to the second is as short as possible. If more than one string contends for this distinction, choose the earliest occurring one in W.

We shall argue now that $\sigma = \tau$, in particular, σ is the initial string of W. Note that there must be a distinct instance of σ_- between σ and its next occurrence. In other words W must contain

$$\sigma \ldots x\sigma_- \ldots \sigma$$

for some binary digit x. (As in [7], we understand in this and similar diagrams that the indicated substrings are distinct and occur in the indicated order, but that overlap may occur.) If the predecessors of the two σs were the same, then the first one would have to be the initial string of W, since σ was chosen to occur as early as possible in W. Thus we may assume the two σs have opposite predecessors. In that case, one of the two predecessors must be x. If it were the predecessor of the second σ then the substring $x\sigma_- \ldots x\sigma_-$ would be shorter than $\sigma \ldots \sigma$, contrary to assumption. Thus x must match the predecessor of the first σ. But in that case we reach a similar contradiction unless the first σ were the initial string of W.

By the forgoing, every string of length $n+1$ that occurs in W must recur at equally spaced intervals, and that spacing is the same as for σ, the initial string of W. Thus $W = \sigma_{-}Y\sigma_{-}Y\ldots$ for some string Y. Let $X = \sigma_{-}Y\sigma_{-}$. Let μ be any given string of length $n+1$. By Proposition 3.2, some instance of μ must begin in $\sigma_{-}Y$. Thus, μ is a substring of X. It follows that X contains every given string of length $n+1$ exactly once, and it is therefore a DeBruijn string. □

The previous result is impractical for generating DeBruijn sequences from an arbitrary seed since it gives no upper bound on the number of steps required. On the other hand, it is easy to produce seed strings Z for which $EM_n(Z)$ is immediately periodic.

Theorem 3.2 Let Z be any string ending in n zeros that contains every n-string at least once. Then the initial 2^{n+1}-string of $EM_n(Z)$, together with the n zeros at the end of Z, is a DeBruijn string of order $n+1$.

Proof: [3] presents an algorithm that produces an Eulerian circuit of $\mathcal{B}^n$ from a directed tree in $\mathcal{B}^n$ rooted at $\bar{0}_n$.(The corresponding Hamilitonian circuit of $\mathcal{B}^{n+1}$ is then a DeBruijn string of order $n+1$.) Consider the subgraph, T, of $\mathcal{B}^n$ formed by connecting the vertex of an n-string σ to the vertex of the n-string that follows the *last* occurrence of σ in Z. It is obvious that every n-string can be connected to $\bar{0}_n$: just follow along Z and jump to the last occurrence whenever necessary. Thus T is a tree rooted at $\bar{0}_n$. Now apply ALGORITHM 1 on page 200 of [3]. (One must replace the n in [3] with $n+1$.) It is only necessary to note that the steps of the algorithm are identical with those of $EM_n(Z)$. □

With a slight modification, the algorithm can be adapted to handle arbitrary seeds: Start with $\bar{0}_n$. Every time an n-string tail is encountered that does not appear in the seed, pre-pend it to the beginning of the seed and continue as if the new longer seed had been the one given. The algorithm is very fast, and is simple enough to use by hand. For example, with $n = 3$ and an empty seed, it produces the order 4 DeBruijn string 0001111010110010000. (In the course of the algorithm, the original empty seed turns into 111011001000.)

We return now to the study of the usual EM sequence (i.e., with h equal to the identity function,) but with a nonempty seed string Z. The resulting sequences EM(Z) share some of the important features of the usual EM sequence. For example, we have

Proposition 3.4 Let m_0 be the length of the longest string that recurs in Z. Let σ be the match string of EM(Z) at time n. If $|\sigma| > m_0$ and the match length m_n reaches a new record value at time $n+1$, i.e., $m_{n+1} > m_k, k = 1, 2, \ldots, n$, then σ is a head of ZEM(Z).

The proof is essentially the same as that of Proposition 4.1 of [7].

Let T_n be the first time the match length reaches n. Then we have

$$T_n \leq 2^n + n - |Z| < 2^{n+1} - |Z|, n > m_0. \tag{3.2}$$

To see this, note that if $T_n > k$ then the finite string $ZEM(Z)_k$ contains no repeated strings of length n. Thus $k+|Z|-n+1 \leq 2^n$ and the desired inequality follows by taking $k = T_n - 1$.

Like EM, the strings $EM(Z)$ are always transitive. Let C_n be the *cover time* of strings of length n, i.e., the smallest k for which every string of length n is a substring of $EM(Z)_k$. We shall show $EM(Z)$ is transitive by obtaining a crude upper bound for the C_n. For a given positive constant C, define a function f on the natural numbers inductively by setting $f(1) = C$, and $f(n+1) = 2^{f(n)+|Z|+2}$.

Proposition 3.5 The constant C can be chosen depending only on Z such that $C_n \leq f(n)$.

Proof: We proceed by induction on n. It is easy to see that EM(Z) cannot consist entirely of ones or entirely of zeros. Thus we can handle the case $n = 1$ merely by taking the constant C sufficiently large.

Put $N = f(n)$ and assume as an inductive hypothesis that every string τ of length n occurs at least once in $EM(Z)_N$. Fix any such τ and let σ be the prefix of ZEM(Z) that ends at one of the occurrences of τ in $EM(Z)_N$. By Proposition 3.4, both $\sigma 0$ and $\sigma 1$ occur in the initial segment of ZEM(Z) having length $T_{|\sigma|+1}$. Hence, both $\tau 0$ and $\tau 1$ occur in the initial segment of EM(Z) having length $T_{|\sigma|+1} - |Z|$. But $|\sigma| \leq N + |Z|$, so by (3.2) $T_{|\sigma|+1} \leq 2^{|\sigma|+2} \leq 2^{N+|Z|+2} = f(n+1)$.

Since τ was arbitrary, every string of length $n+1$ occurs at least once in the initial segment of EM(Z) of length $f(n+1)$, i.e., $C_{n+1} \leq f(n+1)$. This completes the inductive step, and the proof.

Next, we show that Z is uniquely determined by EM(Z).

Theorem 3.3 Let Z_1 and Z_2 be binary strings. Then if $EM(Z_1) = Y = EM(Z_2)$, the strings Z_1 and Z_2 are equal.

Proof: Assume $Z_1 \neq Z_2$. We may assume that $|Z_1| \geq |Z_2|$ and that Z_1 is not the empty string. Let $X_1 = Z_1 EM(Z_1)$ and $X_2 = Z_2 EM(Z_2)$.

Let σ be the head of Y of length k. Since Y is not periodic, the distance $d(k)$ between the first two occurrences of σ in Y tends to infinity as k tends to infinity. Fix k with $d(k) \geq |Z_1| + 1$.

The first matches of $\_Z_1\sigma$ and $Z_1\sigma$ in X_1 must follow one of the following two patterns:

i

$$Z_1\sigma x \quad \ldots \quad \_Z_1\sigma x' \quad \ldots \quad Z_1\sigma x'$$

ii

$$Z_1\sigma xy \quad \ldots \quad Z_1\sigma x' \quad \ldots \quad \_Z_1\sigma xy'$$

See, e.g., Theorem 4.6 of [7] or Lemma 2 of [9]. One may check that these results generalize to the seeded case provided all match lengths involved in the arguments exceed the length of the seed. (In the language of [7], an excursion begins in case (ii) at the middle sequence in the display, and ends at the last sequence. The key point here is that the match of $\_Z_1\sigma$ occurs before the next appearance of $Z_1\sigma$.)

In case (i), the choice of k ensures that the string denoted $\_Z_1\sigma x' \ldots Z_1\sigma x'$ lies entirely inside Y, i.e., does not overlap with the seed. Thus, in X_2 we have

$$Z_2\sigma x \quad \ldots \quad \_Z_1\sigma x' \quad \ldots \quad Z_1\sigma$$

Since $|Z_1| \geq |Z_2|$ and Z_1 and Z_2 are not equal, there can be no other occurrence of $Z_1\sigma$ in the range shown. But then the match of $\_Z_1\sigma$ at the end would produce a next digit of x rather than x', a contradiction.

In case (ii), we replace σ with the next longer initial string σx. Then at the next occurrence of $Z_1\sigma x$ in X_1 we are back in case (i). □

4 Double Helices

A given DeBruijn string X of order n induces a Hamiltonian circuit of the DeBruijn graph $\mathcal{B}^n$ that begins at the node corresponding to the initial n-string of X. For example, the order 3 string 0011101000 induces the circuit 001 → 011 → 111 → 110 → 101 → 010 → 100 → 000 → 001 of $\mathcal{B}^3$. If the edges of such a circuit are removed from $\mathcal{B}^n$, the remaining graph contains at least 3 connected components, including the loops from the all zero n-string back to itself, and from the all 1 n-string back to itself. For the order 3 example just given, there is a single additional component consisting of the loop 001 → 010 → 101 → 011 → 110 → 100 → 001.

In cases where there are exactly 3 connected components (i.e., the minimum number possible,) we shall call the original DeBruijn string X a *double helix.* The long cycle in the graph formed by removing the edges of the path of X from the DeBruijn graph will be called the *message loop of X,* and the corresponding string the *message of X.* We also call a DeBruijn cycle a double helix if the associated DeBruijn string is a double helix.

The message of an order n double helix is not itself a DeBruijn string since it does not include the all-zero and all-one strings, $\bar{0}_n$ and $\bar{1}_n$. It does contain every other binary string of length n. Such a string can be converted to a *bona fide* DeBruijn string by inserting an extra 0 at the position of $\bar{0}_{n-1}$ and an extra 1 at the position of $\bar{1}_{n-1}$. We shall call a binary string that contains every string of length n exactly once, except for one or both of the all-zero and all-one strings, a *depleted DeBruijn string.*

Every DeBruijn string of order 3 or less is a double helix, but this is not true for higher orders. For example, 1111000010011010111 is a double helix of order 4, but the DeBruijn string 1111000011010010111 is not a double helix.

If a double helix Z of order n is used as a seed for either EM or EM_n, then it is nearly correct to say that the seeded sequence begins by extracting the message of Z. More precisely, we have:

Theorem 4.1 Let Z be an order n double helix whose initial $n-$string τ is neither the all-zero nor the all-one string. Let Y be the initial $2^n - 1$ string of either $EM_n(Z)$ or $EM(Z)$. Then the path of $\tau_{-}Y$ is the message loop of Z, and ZY is a depleted DeBruijn string of order $n+1$.

Proof: The theorem is vacuous if $n = 1$ and is readily checked for $n = 2$, so we may assume that $n \geq 3$. Let σxy, where x and y are binary digits, be the initial $n+1$ string of Z. Then σ is also the terminal $n-1$ string of Z, and xy' are the first two digits of Y. Now $\tau = \sigma x$ lies on the message loop L of Z in $\mathcal{B}^n$, and $\sigma x \to {}_{-}\sigma xy'$ is an edge of this loop. By induction, each successive edge of the path of ZY is an edge of L, until the entire loop shall have been traversed. Completion of L starting from σx with a return to σx on the last step requires $2^n - 1$ steps, since the path visits every $n-$string except $\bar{0}_n$ and $\bar{1}_n$ exactly once.

Every $n+1$ string except $\bar{0}_{n+1}, \bar{1}_{n+1}$ occurs exactly once in ZY. Therefore, ZY is a depleted DeBruijn string of order $n+1$. □

The method of Theorem 4.1 produces depleted DeBruijn strings of a special type: their first half is a DeBruijn string of the next lower order. They are not themselves depleted double helices, in general.

Double helices do not appear to be particularly rare among DeBruijn strings. For example, 4 of the 16 order 4 DeBruijn cycles are double helices, and 840 of the 2048 order 5 DeBruijn cycles are double helices. We shall show there are double helices of all orders by showing that all DeBruijn strings generated by means of linear recurrences are double helices, but this only accounts for 5 of the 840 double helices of order 5. It would be of interest to find other algorithms that reliably generate double helices of all orders, as well as to count the exact number occurring at each order.

Consider a linear recurrence of the form

$$a_k = c_1 a_{k-1} + c_2 a_{k-2} + \cdots + c_n a_{k-n}, \quad k = 0, 1, \ldots, \tag{4.1}$$

where all operations and elements are those of $\mathbb{Z}_2$, the field of integers modulo 2. If a set of n coefficients c_i is given, as well as a choice of initial values $a_{-1}, a_{-2}, \ldots a_{-n}$, then the recurrence determines a unique infinite sequence of binary digits $a_0, a_1, \ldots$ Since there are 2^n possible $n-$tuples of binary digits, the sequence must eventually become periodic with period at most 2^n. The maximal period is, in fact, $2^n - 1$, since a sequence of n zero digits can only be followed by zero digits.

Binary sequences produced by a linear recurrence that have maximal period are called PN sequences in [4]. A full period of a PN sequence contains all binary sequences of length n except for the all zero sequence. Inserting an extra zero at the beginning of the unique sequence of $n-1$ zeros in such a cycle produces a DeBruijn cycle. We shall call these cycles and their associated sequences *linear DeBruijn* cycles and sequences.

Clearly a necessary condition for a PN sequence is that coefficient c_n should be non-zero. It is convenient to introduce an additional coefficient $c_0 = 1$, and to define the *characteristic polynomial*

$$f(x) = \sum_{j=0}^{n} c_j x^j.$$

It is shown in [4] that characteristic polynomials of PN sequences are necessarily irreducible in the polynomial ring $\mathbb{Z}_2[x]$, and that the period p of any sequence generated by a linear recurrence, for which the characteristic polynomial is irreducible, is determined as the smallest value of p such that the polynomial $1 - x^p$ is divisible by the characteristic polynomial. Furthermore, PN sequences exist for every order n. It therefore follows from our next result that double helices exist for all orders n.

Theorem 4.2 Every linear DeBruijn sequence is a double helix.

Proof: Let a_n be a PN sequence determined by the recurrence (4.1) and having characteristic function $f(x)$. Let b_n be the sequence determined by the linear recurrence

$$b_k = c_1 b_{k-1} + c_2 b_{k-2} + \cdots + c_n b_{k-n} + 1, \quad k = 0, 1, \ldots, \tag{4.2}$$

with initial condition $b_{-1} = 1, b_{-2} = b_{-3} = \cdots = b_{-n} = 0$. The paths in $\mathcal{B}^n$ induced by the a_k sequence and the b_k sequence leave each vertex by the opposite edge, so it suffices to show that the period of the latter sequence is $q = 2^n - 1$.

Let $G(x)$ be the generating function of the b_k sequence, i.e.,

$$G(x) = \sum_{k=0}^{\infty} b_k x^k.$$

Substituting for b_k the expression on the right side of (4.2) and interchanging orders of summation, we obtain

$$G(x) = \frac{1}{1-x} + \sum_{i=1}^{n} c_i x^i (b_{-i} x^{-i} + \cdots + b_{-1} x^{-1} + G(x)),$$

whence

$$G(x) f(x) = \frac{1}{1-x} + \frac{1 - f(x)}{x} = \frac{1 + x f(x) + f(x)}{x(1-x)}.$$

(Recall that operations are done in $\mathbb{Z}_2[x]$.) Now since $f(0) = 1$, the last written numerator is divisible by x, i.e.

$$h(x) = \frac{1 + x f(x) + f(x)}{x} = f(x) + \sum_{i=1}^{n} c_i x^{i-1} \tag{4.3}$$

is a polynomial that also satisfies

$$G(x)(1-x) f(x) = h(x).$$

On the other hand, since b_k has period q, we have

$$G(x) = \frac{g(x)}{1 - x^q}$$

for some polynomial $g(x)$. Combining these results we have $g(x)(1-x)f(x) = h(x)(1-x^q)$. Since $f(x)$ is irreducible, it must divide either $h(x)$ or $1-x^q$. But by (4.3) the first possibility cannot occur, since then f would divide a polynomial of lower degree. It follows that f divides $1-x^q$, and hence $q \geq 2^n - 1$. But since the b_n sequence never visits the all one n-string, its period is exactly $2^n - 1$, as required. (Since a PN sequence does visit $\bar{1}_n$, it is necessary that $c_1 + c_2 + \cdots + c_n = 0$ to avoid getting trapped there. Thus the b_n sequence cannot visit $\bar{1}_n$ unless it starts there.) The path induced in $\mathcal{B}^n$ by the b_n sequence takes the form $\bar{0}_{n-1}1 \to \cdots \to 1\bar{0}_{n-1} \to \bar{0}_n \to \bar{0}_{n-1}1$. ($2^n - 1$ steps.) If the original PN sequence is converted to a DeBruijn sequence by inserting a zero, then in the message loop the last two steps are replaced by the single step $1\bar{0}_{n-1} \to \bar{0}_{n-1}1$. Thus the message loop omits both $\bar{0}_n$ and $\bar{1}_n$. □

References

[1] A. Ehrenfeucht and J. Mycielski, *A pseudo-random sequence - how random is it?*, Amer. Math. Monthly **99** (1992), 247-271.

[2] http://barnyard.syr.edu/mseq/mseq.1000k.gz (gzipped ascii file.)

[3] H. Fredricksen,*A Survey of Full Length Nonlinear Shift Register Cycle Algorithms*, SIAM Review **24**(1982), 195-221.

[4] S. W. Golomb, *Shift Register Sequences*, Aegean Park Press, Willow Creek, California, 1982.

[5] Grzegorcz Herman and Michael Soltys, *On the Ehrenfeucht-Mycielski sequence,* Journal of Discrete Algorithms, **7**(2009),500-508.

[6] John C. Kieffer and W. Spankowski, *On the Ehrenfeucht-Mycielski balance conjecture*, DMTCS Proceedings from the Conference on Analysis of Algorithms, (2007),19-28.

[7] T.R. McConnell, Laws of Large Numbers for some Non-repetitive Sequences, unpublished manuscript.

[8] http://oeis.org/

[9] Klaus Sutner, *The Ehrenfeucht-Mycielski sequence"*, Lecture Notes in Computer Science, **2759**(2003), 282-293.